

*Amendments to the Claims*

1-4. (canceled)

5. (currently amended)      A method for presenting document records to a user through a display interface, comprising the steps of:

- (a)      managing a plurality of data files with a host application, the host application supporting applet execution;
- (b)      selecting a data file from a plurality of data files;
- (c)      analyzing the contents of the data file for the presence of data of a first data type and a second data type;
- (d)      processing data of the first data type through a first applet and data of the second data type through a second applet, wherein the first applet is a first reader for data of the first data type, and the second applet is a second reader for data of the second data type that reads data from the data file after a header of the data file until an end of file marker is encountered in the data file, wherein data of the first data type is located in the data file after the end of file marker;
- (e)      merging and formatting the processed first and second data within the host application; and
- (f)      displaying the merged and formatted processed first and second data.

6. (previously presented)      The method according to claim 5, wherein the first data type is a graphics type and the second data type is a text data type.

7. (previously presented)      The method according to claim 5, wherein the data file comprises a tagged format.

8. (previously presented)      The method according to claim 5, wherein the first data type comprises a compressed format image.

9. (previously presented)      The method according to claim 5, wherein the data file comprises:

    a header portion containing an index portion;

    a first data type located near a terminus of the data file at a starting location referenced by the index portion; and

    a second data type located between the header and the first data type, having an end of file marker at its terminus.

10. (previously presented)      The method according to claim 5, wherein the host application comprises a hypertext browser.

11-13. (canceled)

14. (currently amended)      A method of processing a data file having two different data types, comprising the steps of:

(a) analyzing the contents of the data file for data of a first data type and a second data type;

(b) processing the data file with a first applet, adapted for reading data of the first data type, to extract data of the first data type;

(c) processing the data file with a second applet, adapted for reading data of the second data type, to extract data of the second data type, wherein the second applet reads data from the data file after a header of the data file until an end of file marker is encountered in the data file;

wherein the data file includes an index portion in a header pointing to the first data type, and the second data type resides between the header and the first data type, having an end of file marker at a terminus thereof.

15. (previously presented) The method according to claim 14, wherein the first applet skips past the end of file marker based on the index portion, thereby circumventing processing of the second data type.

16. (previously presented) The method according to claim 14, wherein an object browser accesses the data file, and invokes the first and second applets for interpreting the composite data.

17. (previously presented) The method according to claim 14, wherein the first data type is a graphics type and the second data type is a text data type.

18. (previously presented) The method according to claim 14, wherein the header and first data type are compatible with the Group 4 Tagged Image Format File specifications.

19. (previously presented) The method of claim 5, wherein step (b) comprises:  
displaying a list of the plurality of data files; and  
enabling a user to select the data file from the displayed list.

20. (previously presented) The method of claim 19, wherein said enabling step comprises:  
enabling a user to sort the displayed list.

21. (previously presented) The method of claim 5, wherein the host application includes at least one control accessible in a window-based environment Windows Visual Basic (VBX) control, wherein step (f) comprises:

(1) displaying the merged and formatted processed first and second data using the at least one VBX control.

22. (previously presented) The method of claim 21, wherein the at least one VBX control includes an Accusoft ImageGEAR a product control produced by Accusoft, wherein the first data type is a graphics data type, wherein step (1) comprises:

displaying the graphics data type using the Accusoft ImageGEAR product control.

23. (previously presented) The method of claim 21, wherein the at least one VBX control includes a BennetTec AllText product control produced by BennetTec, wherein the second data type is a text data type, wherein step (1) comprises:

displaying the text data type using the BennetTec AllText product control.

24. (previously presented) The method of claim 16, further comprising:

displaying a list of data files at the object browser; and

enabling a user to select the data file from the displayed list.

25. (previously presented) The method of claim 24, wherein said enabling step comprises:

enabling a user to sort the displayed list.

26. (currently amended) A system for presenting document records to a user, comprising:

a display interface; and

a processor, wherein said processor is configured to

enable management of a plurality of data files with a host application, the host application supporting applet execution,

enable selection of a data file from a plurality of data files,

analyze the contents of the data file for the presence of data of a first data type and a second data type,

process data of the first type through a first applet and data of the second type through a second applet, wherein the first applet is a first reader for data of the first data type, and the second applet is a second reader for data of the second data type that reads data from the data file after a header of the data file until an end of file marker is encountered in the data file, wherein data of the first data type is located in the data file after the end of file marker, and

merge and format the processed first and second data within the host application;

provide the merged and formatted processed first and second data to the display interface for display.

27. (previously presented) The system of claim 26, wherein the first data type is a graphics type and the second data type is a text data type.

28. (previously presented) The system of claim 26, wherein said processor is further configured to cause said display interface to display a list of the plurality of data files, and to enable a user to select the data file from the displayed list.

29. (previously presented) The system of claim 28, wherein said processor is further configured to enable a user to sort the displayed list.

30. (currently amended) A system for processing a data file having two different data types, comprising:

a processor configured to

analyze the contents of the data file for data of a first data type and a second data type,

process the data file with a first applet that is adapted for reading data of the first data type, to extract data of the first data type, and

process the data file with a second applet that is adapted for reading data of the second data type, to extract data of the second data type, wherein the second applet reads data from the data file after a header of the data file until an end of file marker is encountered in the data file;

wherein the data file includes an index portion in a header pointing to the first data type, and the second data type resides between the header and the first data type, having an end of file marker at a terminus thereof.

31. (previously presented) The system according to claim 30, wherein the first applet skips past the end of file marker based on the index portion, thereby circumventing processing of the second data type.

32. (previously presented) The system according to claim 30, wherein said processor is further configured to access the data file with an object browser, and to invoke the first and second applets for interpreting the composite data.

33. (previously presented) The system of claim 30, wherein the first data type is a graphics type and the second data type is a text data type.

34. (previously presented) The system of claim 32, further comprising:

a display interface;

wherein said processor is further configured to cause said display interface to display a list of data files at the object browser, and to enable a user to select the data file from the displayed list.

35. (previously presented) The system of claim 34, wherein said processor is further configured to enable a user to sort the displayed list.

36. (currently amended) A method, comprising:

(a) receiving at least one data file in a host application;

(b) processing the contents of the data file for data of a first data type through a first applet and a second data type through a second applet, wherein the first applet is a first reader for data of the first data type, and the second applet is a second reader for data of the second data type that reads data from the data file after a header of the data file until an end of file marker is encountered in the data file, wherein data of the first data type is located in the data file after the end of file marker; and

(c) formatting the processed data of the first and second data types within a host application.

37. (previously presented) The method of claim 36, further comprising:

(d) displaying the captured data.



38. (previously presented) The method of claim 36, further comprising:

(d) merging the processed data of the first and second data types within the host application.

39. (previously presented) The method of claim 36, further comprising:

(d) locating the data of the first data type and the second data type in the data file.

40. (previously presented) The method of claim 36, wherein the first data type is a graphics type and the second data type is a text data type.

41. (previously presented) The method of claim 36, wherein the data file comprises a tagged format.

42. (previously presented) The method of claim 36, wherein the first data type comprises a compressed format image.

43. (previously presented) The method of claim 36, wherein the data file comprises:

a header portion containing an index portion;

a first data type located near a terminus of the data file at a starting location referenced by the index portion; and

a second data type located between the header and the first data type, having an end of file marker at its terminus.

44. (previously presented) The method of claim 36, wherein the host application comprises a hypertext browser.

45. (new) A computer program product comprising a computer usable medium having computer readable program code means embodied in said medium for presenting document records to a user through a display interface, said computer readable program code means comprising:

a first computer readable program code means for enabling a processor to manage a plurality of data files with a host application, the host application supporting applet execution;

a second computer readable program code means for enabling a processor to select a data file from a plurality of data files;

a third computer readable program code means for enabling a processor to analyze the contents of the data file for the presence of data of a first data type and a second data type;

a fourth computer readable program code means for enabling a processor to process data of the first data type through a first applet and data of the second data type through a second applet, wherein the first applet is a first reader for data of the first data type, and the second applet is a second reader for data of the second data type that reads data from the data file after a header of the data file until an end of file marker is

encountered in the data file, wherein data of the first data type is located in the data file after the end of file marker;

a fifth computer readable program code means for enabling a processor to merge and format the processed first and second data within the host application; and

a sixth computer readable program code means for enabling a processor to display the merged and formatted processed first and second data.

46. (new) The computer program product of claim 45, wherein the first data type is a graphics type and the second data type is a text data type.

47. (new) The computer program product of claim 45, wherein the data file comprises a tagged format.

48. (new) The computer program product of claim 45, wherein the first data type comprises a compressed format image.

49. (new) The computer program product of claim 45, wherein the data file comprises:

a header portion containing an index portion;

a first data type located near a terminus of the data file at a starting location referenced by the index portion; and

a second data type located between the header and the first data type, having an end of file marker at its terminus.

50. (new)      The computer program product of claim 45, wherein the host application comprises a hypertext browser.